

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 1/00, 9/44		A1	(11) International Publication Number: WO 99/56196 (43) International Publication Date: 4 November 1999 (04.11.99)
(21) International Application Number: PCT/US99/09622		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 30 April 1999 (30.04.99)		Published <i>With international search report.</i>	
(30) Priority Data: 09/070,617 30 April 1998 (30.04.98) US			
(71) Applicant: BINDVIEW DEVELOPMENT CORPORATION [US/US]; 5151 San Felipe, Houston, TX 77056 (US).			
(72) Inventors: SHOSTACK, Adam; 423 Brookline Avenue, Boston, MA 02215 (US). ALLOUCH, David; 7A Haoranim Street, Givat Shimuel (IL).			
(74) Agent: TOEDT, D., C., III; Arnold, White & Durkee, P.O. Box 4433, Houston, TX 77210 (US).			
(54) Title: COMPUTER SECURITY			
(57) Abstract			
The invention relates to providing enhancements automatically to computer security software whenever the enhancement becomes available. The invention also relates to providing automatically an update to computer security software and integrating the update into the software.			

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Larvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav	TM	Turkmenistan
BF	Burkina Faso	GR	Greece		Republic of Macedonia	TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

COMPUTER SECURITY

This invention relates generally to computer security software and systems.

BACKGROUND INFORMATION

5 The rapid development of intranets, extranets and the internet has introduced an increased level of security problems for network managers, computer information systems professionals, individual users, and corporations with an expanding base of telecommuters. With the advent of electronic mail and electronic commerce via the internet, computer information security is an increasing worldwide concern. The responsibilities of system 10 administrators to provide and monitor network connections for security breaches has substantially increased. Furthermore, with the rapid increase of new computer users and the constant development of sophisticated techniques for breaching established network security systems, system administrators are unable to provide their clients and servers with adequate protection. As a result, computer network systems have become increasingly 15 vulnerable to attacks.

 In an attempt to prevent unwanted access to computer networks, systems administrators have employed various techniques. One such technique employs a firewall to protect the network clients and servers. A firewall is a screen between a user external to the network and the network and is usually the first line of defense against unauthorized 20 users seeking access to a network. The firewall behaves much like an electronic filter that determines whether a particular user has the requisite security clearance to gain access to the network or computer. As an initial defense, the firewall generally provides adequate protection. However, depending upon the concentration of network traffic, quality of the firewall, and the sophistication, skill and motivation of the person seeking access, the 25 firewall becomes vulnerable to attack. Furthermore, firewalls are designed to prevent unauthorized external access and do not prevent internal users from breaching network security.

 In addition, there are products available in the public domain directed to uncovering security vulnerabilities within networks. Although the software tools are not 30 explicitly designed for use by hackers, the tools may be used to gain unauthorized access

to a network. For example, a software tool that is widely available is the system administrator tool for analyzing networks (SATAN). This software tool may be used to probe for security holes within a network and highlight network vulnerabilities. An intruder is then able to take advantage of the information obtained from SATAN to gain 5 unauthorized access to a network.

The list of network vulnerabilities is always changing and usually well known by hackers. Over the years, hackers have developed many techniques for breaching computer security. Many of the techniques often involve exploiting the vulnerabilities associated with particular software packages. For example, hackers are aware of vulnerabilities in 10 software programs like electronic mail (e-mail), software features like remote login(rlogin), or security weaknesses in particular word processing programs, and they use this information to gain unauthorized access to a network or computer.

One technique used by hackers to breach computer network security is Internet Protocol spoofing (IP spoofing). Using this technique, an unauthorized user gains access to 15 a network by hiding their true location and masking their Internet Protocol (IP) address or root address. In doing so, the IP address appears acceptable to a network server and the unauthorized user is granted access to the network.

Another known method for breaching network security is the buffer overflow technique. Hackers use this technique to gain access to a network through insecure 20 implementation of a file in a file transfer protocol (FTP) server, an electronic mail system, a network file server (NFS), or through a common gateway interface (CGI). The buffer is essentially a temporary holding place in memory with a fixed size for processing computer programs and a hacker may cause too much information to be placed in a buffer. When the buffer is beyond its capacity, an overflow occurs. The overflow is then sent to another part 25 of memory within a server. The hacker is then able to gain privileged access to the computer from inside the new location in memory, and as a result, security is breached.

Whenever an unauthorized user breaches network security and is allowed free access to the system, the damage that might result is unpredictable. However, because some of the system vulnerabilities and techniques used by hackers are known, a system 30 administrator may use that information to make the network less vulnerable to attack. However, the system administrator is required to remain constantly vigilant as to the new

attacks being used by hackers, and then use that information to protect the network, clients and servers from the newly found vulnerability.

SUMMARY OF THE INVENTION

While some system administrators may be equipped with software packages that assist them in providing security for their networks, updates to those software packages typically are not automatically provided in real-time, nor are they provided as soon as a new vulnerability is discovered. One aspect of the present invention is that it automatically provides, in real-time, software enhancements with updated information regarding security vulnerabilities. Thus, a user, system administrator, server, etc. is able to implement prevention techniques before a security breach occurs. In accordance with this aspect of the invention, the enhancement that was sent is then integrated into the computer security software. Before the integration, a computer check can be performed to determine the integrity and the authenticity of the enhancement. The computer check can use cryptographic techniques such as digital signatures and Pretty Good Privacy™ (PGP™) encryption.

In one aspect, the invention provides the most recent information regarding new security attacks. A user can either request the enhancement, or it can be automatically sent (e.g., via the internet) when it becomes available. The software enhancement can include a new version of the software and an update to a database of known security vulnerabilities. A user thus can obtain instant access to the latest security vulnerabilities and employ immediate remedial action before a security breach occurs. Thus, systems and methods according to the invention are not bounded by a static database of security vulnerabilities information. The present invention obviates the need to manually update a computer security system.

In another aspect, the invention relates to a network security detector that is used to monitor security intrusions on a network. The network security detector (NSD) may consist of a single software application dedicated to continuously scanning the network. However, in the disclosed invention the NSD consists of a first application that provides real-time intrusion detection; a second application that behaves like a system manager; a third application that is able to simulate attacks on the network and monitor Internet

Protocol devices; a fourth application that performs a comprehensive security assessment of the network; and a fifth application responsible for receiving the software enhancements.

In another aspect, the invention relates to an integrated system for assessing 5 vulnerabilities. The integrated system includes a database of security vulnerabilities and various modules. A first module accesses the database and assesses security vulnerabilities of an operating system of a computer. A second module accesses the database and assesses security vulnerabilities of a computer network that includes the computer. A third module accesses the database and assesses security vulnerabilities in passwords used to access the 10 computer or the network. A fourth module accesses the database and assesses security vulnerabilities of a remote computer connected to the network. A fifth module receives an update to the database and updates the database. A sixth module is a communications module that allows communication between the integrated system and a similar system.

The foregoing and other objects, aspects, features, and advantages of the invention 15 will become more apparent from the following description and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings, like reference characters generally refer to the same parts throughout the different views. Also, the drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the invention.

20 FIG. 1 is a schematic diagram of a computer network.

FIG. 2 is a schematic diagram of the network security detector for providing security on a local area network.

FIG. 3 is a schematic diagram of a "push" system for delivering enhancements to a computer security system.

25 FIG. 4A-1 and 4A-2 represent a flow chart of a procedure for installing enhancements to a security vulnerabilities database.

FIG. 4B is a flow chart of a procedure for extracting the filename of a security vulnerabilities database.

30 FIG. 5 is schematic diagram of the integrated security system modules for assessing security vulnerabilities on a computer network.

FIG. 6 is a graphical user interface that shows some of the functions presented to a user on a computer monitor, each of the functions relating to assessing security vulnerabilities on a computer and/or a computer network.

FIG. 7 is a flow chart of a Pretty Good Privacy™ (PGP™) authentication procedure for checking the integrity and authenticity of a software enhancement or update.

DESCRIPTION

In accordance with the invention, a database of security vulnerabilities is automatically updated via an electronic network. The database is part of a computer security software system. The automatic update can occur whenever a software enhancement becomes available. The update can then be integrated into the computer security software. New and different security vulnerabilities are discovered almost daily. As a result, computer security checks should employ a flexible mechanism able to adapt to newly discovered security vulnerabilities. The present invention provides such a mechanism by automatically providing enhancements to a database of security vulnerabilities and using that information to provide security solutions to potentially "weak" computer networks and/or computers.

Referring to FIG. 1, a network 20 includes a local network environment 10, a data bus 14 for electronically linking various ports of the network, a local server 18, a network security detector (NSD) 16, and a firewall 12 for screening unauthorized external users 8. The local network environment 10 includes connections to the internet and routers for connecting authorized remote locations. System administrators can create a secure environment by using the firewall 12 and supplementing it with the network security detector 16.

The firewall 12 is an electronic filter used to prevent unauthorized external users 8 from accessing the network 20 without permission. The network security detector 16 ascertains whether unauthorized external users 8 and authorized local users 6 possess the requisite security clearance to access certain areas within the network 20. The network security detector 16 can be used to prevent both unauthorized external users 8 and authorized local users 6 from unauthorized access to the local server 18. The firewall 12 and/or the NSD 16 may be subject to attack.

The network security detector 16 can be electrically connected to a database of security vulnerabilities which may be stored on the local server 18. However, in another embodiment of the invention, a database of security vulnerabilities is stored on the individual computers of the authorized local users 6 or at a remote location not local to the network 20. The database of security vulnerabilities includes a list of techniques used by hackers to gain unauthorized access to the network 20 and includes a catalog of known security weaknesses in software programs stored on the network 20. In one embodiment of the invention, the database of security vulnerabilities is used in conjunction with the network security detector 16 to provide security for the network 20.

As previously mentioned, the list of computer and network vulnerabilities is always changing and growing, and over the years, hackers have developed many techniques for breaching computer security. Table 1 is a list of some of the known features that hackers have used to gain access to computer networks. Table 1 lists common vulnerabilities found on the network 20 that might be exploited. Although the firewall 12 is normally the first line of defense against an attack on the network 20, the firewall 12 can be circumvented using many techniques, such as IP spoofing and other types of attacks discussed below.

Table 1 shows that features often provided for the convenience of authorized network users may be exploited by hackers or unauthorized external users 8 to access the network 20. For example, in UNIX™, *finger* is a feature that allows authorized local users 6 to locate other users on the network 20, or *netstat* is used to obtain information regarding network status. However, an individual motivated to breach network security may use these features to gather information about valid users (such as internet addresses) and then use that information to gain unauthorized access to the network 20. Hackers may also gain unauthorized access by using programs stored on the network 20 like *sendmail* and *X-windows™* that may allow access to program libraries or give out too much information about authorized network users.

Another avenue onto the network 20 is through a daemon which is a program intended to provide useful services that is not explicitly invoked but lies dormant waiting for some condition on the network 20 to occur. The idea is that the perpetrator of the condition need not be aware that a daemon is lurking although a program may commit an

action only to invoke the daemon. For example, printing a file may first invoke the daemon for spooling and then print the file. Another daemon is the hypertext transfer protocol daemon (HTTPD) which is a program used to provide information for the world wide web. However, unless configured properly, the path through an HTTPD may also 5 allow unauthorized external users 8 through the firewall 12 and onto the network 20.

Depending upon the motivation of the unauthorized external user 8, the attack may consist of placing a disruption on the system that limits access and not necessarily removing or copying secret documents. Methods for limiting processing on the network 20 are called denial of service attacks. Some techniques used to deny service is called the 10 teardrop and land attack where a hacker sends pairs of carefully constructed IP fragments to a network server. The IP fragments trigger bugs in computer programs or network logic. The overlapping offsets cause the second packet to overwrite data in the middle of the user datagram protocol (UDP) header in such a way that the datagrams are left incomplete. When a software program then reads the invalid datagrams, the program allocates kernel 15 memory, and if enough of the invalid datagrams are received, then the software program is indefinitely suspended.

There are several other commonly used attacks that produce the aforementioned result namely UDP bombs, ping floods, and SYN floods. SYN represents the synchronizing bit that indicates to a server that a client is seeking access. The SYN flood 20 attack bombards a system with dozens of falsified connection requests a minute and can seriously degrade a system's ability to give service to legitimate connection requests. Accordingly, the attack is said to "deny service" to system users. In addition, a hacker may use some of the preceding techniques to store programs on the network 20 that could be used to gain access at a later time. That is, the hacker builds a backdoor or trap door onto 25 the network 20 that might go undetected, and as a result this enables the hacker to exploit the network 20 at any given time.

TABLE 1: An example of the type of information contained in a security vulnerabilities database.

<u>Feature</u>	<u>Vulnerability</u>
FirewallCheck	the firewall for vulnerability to routing, IP spoofing, and other attacks.
Information Gathering	Check for finger, rusers, netstat, and many other sources of network information that is useful during an attack.
Sendmail	Check for historical vulnerabilities in sendmail and misconfigurations that may give out too much information or allow outsiders in. Also check for buffer overflows that allow local users to gain root.
File Transfer Protocol (FTP)	Check for world-writable directories, insecure files inside the FTP sandbox, and other attacks that may crash the server, plant backdoors into the system, or allow users to escape the FTP sandbox.
Network File Server	Check for insecure shares, filehandle guessing, and other common misconfigurations that allow outsiders to see the disks or create backdoors to the system.
HyperText Transfer Protocol Daemon and Internet Information Server (HTTPD and IIS)	Check for known bugs in the servers, commonly present CGI scripts that are vulnerable to buffer overflow attacks, server misconfigurations, and other vulnerabilities that allow outsiders to escape the server sandbox, gain root or user access to the system, or crash the server or system.
Miscellaneous Daemon	Check for the presence of network services that are inherently insecure, such as telnetd, walld, tfpd, and many others.
X-windows™	Check for open permissions that allow snooping of remote X sessions, unpatched libraries and executables vulnerable to buffer overflow attacks, and other well known X vulnerabilities.
Denial of Service Attacks	Check for vulnerability to all common attacks SYN Floods, UDP Bombs, Ping Floods, and others, plus newer attacks like Land and Teardrop.
Home	Checks home directory for world writeability.
System	Checks for read/write files in the configuration files.
Password	Checks for/etc/passwd, easy guessable passwords.

<u>Feature</u>	<u>Vulnerability</u>
Configuration Files Check	Look for common misconfigurations in various files like inetc.conf, .rhosts, host.equiv, and many others.
Users	Checks for home directory important files with insecure permissions and pays special attention to root's files.
Backdoors	Checks for possible backdoors in the system binaries and configuration files.
Patches	Checks for the presence of all Sun™ security.

The Network Security Detector (NSD)

As previously mentioned, the firewall 12 is often the first line of defense against some of the aforementioned security attacks on the network 20. However, according to the 5 invention, another layer of defense involves the use of a security network detector 16. In one embodiment of the invention, the network security detector 16 includes software programs that seek to uncover security intrusions. In one embodiment of the invention, the network security detector 16 is a single package that continuously scans the network for violators. In another embodiment of the invention, the network security detector 16 is an 10 integrated family of software packages that individually resolve various security issues. Referring to FIG. 2, the network security detector 16 has various components. In one embodiment of the network security detector 16, it has at least four integrated software applications for providing network security.

A first application 48 of the NSD 16 provides a real-time intrusion detection 15 notification system. In one embodiment, the first application 48 takes an action which may include sending an alarm to a system administrator if an intrusion is detected. In addition, the first application 48 electronically disengages the intruder and marks the intruder's location. The first application 48 can also distribute to each computer on the network 20 information about network status. The first application 48 can watch each site location and 20 electronically communicate with a system manager, or his programmatic agent. The system manager can be a system administrator. However, the system manager could be another software program in electrical communication with the first application 48.

A second application 42 of the NSD 16 can include the system manager for receiving information from the first application 48. In this capacity, the system manager assists the system administrator in providing network security information and assists in managing the security of numerous port connections associated with the network 20.

5 A third application 44 of the NSD 16 can provide continuous monitoring of the complete network 20. The third application 44 is in electronic communication with the other applications and may be used as a system management tool. However, in the disclosed embodiment, the third application 44 is used to monitor Internet Protocol devices.

10 In another aspect of the invention, the third application 44 can also simulate an attack on the network 20. The simulation can provide information to the network security detector 16 for uncovering potential security vulnerabilities before the vulnerabilities are exploited. The third application 44 may also check the local server 18 for security vulnerabilities. In addition, the third application 44 provides a map of all ports on the 15 network 20 and pings all Internet Protocol devices to expose potential security vulnerabilities.

20 A fourth application 46 of the NSD 16 can perform a comprehensive security assessment of the network 20. The fourth application assesses the operating system of various computers and monitors the network for security vulnerabilities. The fourth application 46 can also provide a report of all security breaches and provide an appropriate solution based on a database of known security vulnerabilities similar to Table 1. However, because vulnerabilities are constantly changing and new ones are being discovered, an efficient means is required to update the database of security vulnerabilities.

25 A fifth application is responsible for receiving the software enhancements and updating of the database of security vulnerabilities. The following is a discussion of how enhancements are automatically provided immediately and in real-time for a computer security vulnerabilities database.

The Push System

30 Because network security vulnerabilities are constantly changing and new ones being developed by hackers, a system administrator is required to remain vigilant in order to protect the computer network, clients, and servers from the new system vulnerabilities.

The disclosed invention provides a means for obtaining real-time enhancements to a database of security vulnerabilities. In this way, a system administrator or a dedicated network server is able to take immediate action to protect the computer network, clients and servers before a breach occurs. The push system is a method for automatically 5 providing software enhancements in real-time, and is an efficient means for providing enhancements to a database of security vulnerabilities. As described below, the push system is a part of an integrated security system that primarily provides a secure network operating environment. Specifically, the push system is part of a fifth module that receives an update and updates a database of security vulnerabilities.

10 In one embodiment of the invention, the push system provides computer security software enhancements for execution on at least one computer. The push system automatically implements and electronically sends computer software enhancements over a computer network when the software enhancement becomes available. The software enhancement can include an update to a computer security vulnerabilities database or a 15 new version of an entire computer security software package. In either embodiment of the invention, the software enhancement is automatically distributed over an electronic network.

15 In an alternative embodiment, the push system is manually activated by a user seeking an update. In this alternate embodiment, the user is able to send a query to a server 20 about the availability of an enhancement which can include an update of the database or a new version. If the enhancement is available, the server either pushes the enhancement over the network to the user or provides a negative response if it cannot push the enhancement for some reason.

25 The software update can include a combination of old and new information regarding computer security vulnerabilities for inclusion in the database, where succeeding updates primarily add new information to the database. As a result, the information in the database continuously builds. However, if the enhancement is a new version of the computer security software, the new version includes not only the new database of information but also includes new features or functionalities not a part of the original 30 software. Thus, a new version might require overwriting the old database of information or discarding the old version and re-installing a new database.

The push system integrates the software enhancement into existing programs. Additionally, the integration can also perform a check on the integrity and authenticity of the software enhancement provided. This feature determines whether the user being sent the software enhancement is eligible, and checks the integrity and authenticity of the 5 software enhancement. In determining the integrity and authenticity of the software enhancement, the push system can use digital signatures or other cryptographic techniques. In the disclosed embodiment, digital signatures are used to encode the software enhancement by using a signing key, and an authorized local user 6 or customer possesses the correct key for validating the original message.

10 The push system also installs the software enhancements. The installation includes performing a check on the software enhancement and determining the integrity and authenticity of the software enhancement. Referring to FIG. 3, the push system 60 includes three primary components: an update production process 50; a push mechanism 52; and an update processor 54. The push system 60 also includes a customer database 56 and an 15 installer 58. The following is a description of each component.

The Push Mechanism

Still referring to FIG. 3, the push mechanism 52 delivers the software enhancement to the customer and invokes an installer 58 via the update processor 54. The push mechanism 52 delivers the software enhancement using electronic mail to a small script 20 which places the contents of the push onto a storage device 62. The script is a program written in a high level computer language that a local server 18 may execute. The script can include different commands and subroutines for accessing software applications from various memory locations within the computer. The script may be used to implement 25 storing an update of a database of security vulnerabilities 92 on a storage device 62 and then automatically running an update installation procedure 100.

In the disclosed embodiment, the push mechanism 52 is invoked using an electronic mail message system that is delivered using simple mail transfer protocol (SMTP) which is a standard method for transmitting electronic mail to and from the internet. However, the push mechanism may also use a post office protocol (POP) mail 30 server. As is well known in the art, the POP server behaves much like a post office box. The POP server holds mail for the user, and when the user connects to the POP server,

their mail is automatically transferred to them. Thus, in this particular example, the authorized local user 6 could connect to the POP server and the security vulnerabilities enhancement would automatically be pushed onto his computer, server or network.

In another embodiment of the invention, the security vulnerabilities database is delivered using an internet message access protocol (IMAP) mail service. IMAP offers a built-in flexibility that enables an authorized local user 6 to access electronic mail messages from a stand alone computer, workstation, or a laptop computer without transferring files between the computers. Thus a client is able to access remote messages as if they were on a local server 18.

10 Techniques For Implementing Data Transfer

In a client-server configuration of the network 20, the software enhancements are stored on a server, and the server is able to distribute the software enhancements to a client using a variety of techniques. In the disclosed invention, the server is remotely connected to the client's network 20. Using the push mechanism 52 as described above, the remote server is able to initiate contact with a client. Client information is obtained from the customer database 56. When the software enhancements become available, the push mechanism 52 is invoked. The push mechanism 52 takes the contents of a specified location within the remote server and sends the software enhancement to the client via electronic mail. As a result, whenever the software enhancement becomes available, it is immediately pushed over a computer network to the client.

In another aspect of the invention, the client may initiate contact with the remote server by first inquiring whether the software enhancement is available. If the software enhancement is available, that is if there is new information at a specified memory location, then the client is able to receive delivery of the software enhancement. As a result, the client has performed a pull from the server in order to obtain the software enhancements. In another aspect of the invention, the client or another machine acting on behalf of the client may constantly interrogate the remote server about the status of a software enhancement. Similarly, when the software enhancement becomes available it is immediately delivered to the client.

30 In another embodiment of the invention, the software enhancements are provided using a file transfer protocol (FTP) program. As is well known in the art, FTP allows the

direct transfer of files across a computer network. However, unlike the push or pull mechanisms described above, performing an FTP may involve a variety of proxies to facilitate the file transfer. In addition, FTP as a push mechanism would require allowing access to the portion of memory where the software enhancement transfer files are located

5 which would create a network security vulnerability.

In another aspect of the invention, the software enhancements may be automatically sent to an authorized local user 6 or client on a diskette or on a compact disk read only memory (CD-ROM) storage device. However, this technique minimizes being able to immediately obtain enhancements when they become available, and presents

10 logistical problems that should be avoided.

The Update Production Process

In the disclosed invention, the update production process 50 is dedicated to tracking vulnerabilities and maintaining a database of security vulnerabilities. In one aspect of the invention, the update production process 50 identifies and prioritizes the

15 vulnerability, specifies the type of attack, archives the attack source code, creates a report of the vulnerability, and integrates the new attack into a library of known vulnerabilities. The vulnerabilities are tracked from a variety of sources. The sources include mailing lists, internet web sites and information disseminated by hackers. When new vulnerabilities are discovered, they are classified and prioritized. In the disclosed invention, the

20 vulnerabilities are prioritized based on the type of attack using a numerical range from one to ten. The prioritization is based upon the source of the information, the potential damage that the new vulnerability might produce, or the type of attack to which the vulnerability is targeted (e.g., the network, a local attack, or the operating system). The integration of the new vulnerabilities also includes placing the database of known security vulnerabilities in

25 communication with the network security detector 16.

The Installation Procedure

In the disclosed invention, the update processor 54 installs the software enhancement when the enhancement is received by the customer. Referring to FIG. 4A-1, the installation procedure 100 is a means for processing the software enhancement. Note

30 during each step of the installation procedure 100, that if any step fails (Step 108), then the installation procedure 100 stops further installation (Step 150). When the enhancement

arrives at an authorized local user's 6 location, the installation procedure 100 installs the software enhancement onto a storage device 62 and performs a series of reliability checks. Prior to installing the software enhancement on a computer or on a local server 18, the authenticity and integrity of the software enhancement is determined. The authenticity 5 checks may occur either at the user's computer or at the local server 18. The authenticity checks include performing a cryptographic technique by verifying digital signatures, authenticating the software, and verifying the user before installing the software enhancement. The installer 58 is always invoked on a temporary disk file and contains no programming code (e.g., hyper-text markup language (HTML) or channel definition 10 format (CDF), etc.). As a result, this feature allows for optimal push channel independence. That is, because the installer 58 can read and receive the software enhancements in any format and from any location, the installer 58 is push channel independent. The installer 58 is usually invoked on a temporary file which contains no network markup from transport layers, such as HTTP or CDF. This method allows for 15 independence from any particular push channel. Alternatively, details regarding push channel independence may be included as processing information.

The installer 58 begins the installation procedure 100 by first ensuring that the filename of the software enhancement matches a predetermined string (Step 102). In another embodiment of the invention, the filename is represented by a string of variables 20 and numbers appended by a suffix that indicates the type of document. The installer 58 then checks for a specific location to extract the filename (Step 104) to a PGPTM output.

Referring to FIG. 4B, the installer 58 performs a filename extraction sequence 200. The filename extraction sequence 200 consists of determining whether the filename of the software enhancement is at a certain location (Step 202) within the software enhancement. 25 If the filename is at the specified location, then the installer 58 extracts the filename and performs a matching sequence (Step 204). After the filename is extracted and matched, the installer 58 creates an update lock (Step 106). Referring to FIG. 4A-1, the update lock (Step 106) disables any other version of the software enhancement being installed from functioning. The installer 58 then checks for the PGPTM digital signature (Step 110). A 30 digital signature is provided by a software security package residing at a remote location. In the disclosed invention, the remote location is a site residing with the provider of the

software enhancement. The remote location includes a computer that maintains all security keys pertaining to the digital signature. As a precautionary measure, that computer is off-line and electrically isolated from the client-server network. As a result, no new security vulnerabilities are created and an additional level of computer network security is provided.

In one embodiment of the invention, the software security package that provides the digital signature is the network security detector 16. In the disclosed invention, the security software package uses Pretty Good Privacy™ (PGP™) encryption for authentication and provides a digital signature to facilitate authentication. A digital signature is a cryptographic function computed as a message and a user's private key. The private key is a number or a mathematical value that is unique to the sender. The signature function produces a value unique to the private key and the fingerprint value being signed. The private key has a mathematically related public key that anyone may use to verify the signature created by the private key.

The message that is signed is typically a condensed version of the actual message produced by a message digest (MD) or hash algorithm. In general, a message digest algorithm, takes as an input a message of arbitrary length and produces a shorter fingerprint of the input. In the disclosed invention, the message digest algorithm used is called MD5 and produces a 128-bit fingerprint. The message digest algorithm is generated by a transformation function that produces a fixed size representation of the original message. The message digest function has the properties that it is difficult to predict the value of the function for a given input, and that it is difficult to find two arbitrary messages with the same fingerprint, or given a fingerprint, it is difficult to find a second value for the given fingerprint.

At the receiving end of the message, the recipient verifies the signature on the message using the public key. After the encoded message is sent and properly decoded, the PGP™ authentication process (Step 110) is completed. Referring to FIG. 4A-2, the software enhancement package is then sequentially scanned for errors (Step 114-122), and if there is sufficient space (Step 112) on the storage device, the software enhancement is stored on the client's computer, hard drive or server (Step 114).

After storing the software enhancement with the client, the installer 58 updates an archive (Step 118), overwrites any previously existing software enhancement packages (Step 120), and notifies the client or server that the installation procedure 100 is complete (Steps 122). When the installation is complete (Step 116), the installer 58 automatically 5 runs a scan (Step 118) of the network 20 using the newly installed software enhancement to address any new security vulnerabilities uncovered by the installation procedure 100. The update processor 54 also includes solutions for repairing the newly discovered vulnerabilities. The update processor 54 may automatically implement the suggested repairs of the system vulnerabilities and may send a message that the update is completed 10 (Step 122).

In another embodiment of the invention, the installer 58 is able to push source code as a separate enhancement. When the enhancement is received, the source code will not automatically be processed by the update processor 54. However, the update processor 54 may invoke a source code update mechanism that will prompt the user to install the 15 enhancement source code.

An Integrated Security System

The database of security vulnerabilities is part of an integrated system that provides a secure operating environment. The disclosed invention is an integrated system for assessing computer security vulnerabilities. The integrated system includes a database of 20 security vulnerabilities and various modules. A first module accesses the database and assesses security vulnerabilities of an operating system of a computer. A second module accesses the database and assesses security vulnerabilities of a computer network that includes the computer. A third module accesses the database and assesses security vulnerabilities in passwords used to access the computer or the network. A fourth module 25 accesses the database and assesses security vulnerabilities of a remote computer connected to the network. A fifth module receives an update to the database and updates the database. A sixth module is a communications module that allows communication between the integrated security system and a similar system.

Referring to FIG. 5 and 6, the various integrated security system modules 160 are 30 represented by corresponding symbols on a graphical user interface (GUI) screen 70. The first module 74 is used to check the operating system. The check is invoked by using the

check operating systems 74' icon on the GUI screen. The check involves ascertaining whether a user has the correct permission requirements to gain access to the network. Also, in one embodiment of the invention, the first module 74 determines whether all known vulnerabilities have been addressed. Specifically, the first module 74 determines whether 5 the suggested changes resulting from the installation procedure (Step 118) have been made to the operating system.

The first module 78 uses a binary file integrity checking technique using a message digest number 5 (MD5) checksum for files stored on a disk. In the disclosed embodiment of the invention, the checksum is used to verify that no errors have occurred when reading 10 a particular string of bits or a particular file. The checksum value can be any checksum method where it would be difficult to predict the checksum value for a given input. As described below, no two checksums are equivalent, and any changes or corruption of the stored data is detected. Thus, by using a database of MD5 checksum values, the third module 78 is able to determine whether the software enhancement stored on a storage 15 device 62 was modified after a snapshot had been taken, or after the database was created.

The second module 76 accesses the database of security vulnerabilities 92 and assesses network security. The second module 76 connects to a network service, accepts 20 information from the service and interrogates the service. The second module 76 performs a network scan and may be invoked by activating the check network 76' icon on the graphical user interface screen. The network scan produces a map of the network 86 which is essentially an inventory of the Internet Protocol (IP) devices connected to the network. Using network protocol, the integrated system also probes the ports of each of the IP 25 devices for programs that contain security vulnerabilities that may be exploited. The network scan ensures that the network 20 and a local server 18 is protected against any unauthorized access that may penetrate the firewall 12. The network scan for IP devices is invoked using the properties (PROP) icon 72 which enables an authorized local user 6 to configure the various modules.

The third module 78 accesses the database of security vulnerabilities 92 and assess 30 security vulnerabilities in the passwords being used to access a computer or a computer network 20. The third module 78 uses a dictionary of passwords, common English words, and other words to compare and identify vulnerable passwords. The third module may be

invoked by an authorized user 7 by activating the check password 78' icon on the graphical user interface screen. When invoked, the third module 78 checks whether the words in a list have been used as passwords. A stored password is equivalent to an encrypted version of a different password in a dictionary of words or a password list.

5 The fourth module 88 accesses the database of security vulnerabilities 92 and assess the security vulnerabilities of a remote computer connected to the network. The fourth module 88 allows a remote computer to first connect to a network service then accepts information from the service and like the second module 76, also interrogates the service.

10 The fifth module 90 is for receiving an update to the database of security vulnerabilities 92 and updating the database. As described above, the fifth module 90 includes the installer 58. The fifth module 90 checks the authenticity and integrity of the software enhancement. The authenticity of the software enhancement works for either an update or a new version. The authenticity and integrity of the software enhancement is 15 confirmed using the previously described cryptographic methods with PGP™ output from the network security detector 16. In one aspect of the invention, the fifth module 90 also maintains a record of all transactions.

20 A sixth module is a communications module that allows the integrated security system 160 to communicate with a similar system over a computer network. The sixth module may allow communication between the similar system and the various modules and software applications for sharing database files, for sharing workload in breaking long lists of passwords, transmitting reports or data for purposes of analysis, reporting to a management station, configuring files or configuring an operating system, and invoking a 25 remote system to send a software enhancement. The sixth module may also include cryptographic code for protecting the confidentiality and integrity of the information being transmitted.

25 The sixth module may be used for authenticating a user and providing a means for reporting various transactions on the network 20. Specifically, the sixth module may be used to constantly check a user's identification, the integrity of the service connection, and 30 the status of any network processing.

The GUI 70 may also provide a reporting mechanism. The GUI 70 may also include several means for reporting various network transactions. In the disclosed invention, the GUI 70 includes a log view 80 may allow a user to view a text version the update process or log information on a storage device, a log update 82 that generates a 5 report of all security vulnerabilities on the network 20, and a log clear function 84 that allows a user to erase the log.

Referring to FIG. 7, the PGP™ authentication procedure 400 is described. The fifth module 90 performs a message content authentication and verifies that the software enhancement received is exactly the same as the message sent. The fifth module 90 may 10 employ a cryptographic checksum called a message authentication code, or by using digital signatures. The techniques may be used to verify where the message originated, the sender, and the receiver. As a result, the fifth module 90 is able to verify that the actual sender of the message is the person or server that the sender in the message claims to be.

The fifth module 90 uses an asymmetric cryptosystem wherein the recipient of the 15 message is assured the validity of the sender. The fifth module 90 uses a key distribution center or a public key to verify the place of origination. Alternatively, the fifth module 90 may verify a party's identity by using biometrics. In general, biometrics is a method of authenticating a person's identity by using an electronic transmission of personal identifying characteristics of either the recipient, sender, or both.

20 Still referring to FIG. 7, the PGP™ authentication procedure 400 used by the fifth module 90 is described further. A server 401 obtains the most recent software enhancement and seeks to deliver a secure copy of the software enhancement to a customer 414. The server 401 includes an MD5 checksum utility program 404. Using the checksum utility program 404, the software enhancement file is compressed into a 128-bit 25 cryptographic checksum 406 and given an MD5 checksum value 402. For example, an Intel™ system running DOS or Microsoft Windows™ and an executable file "md5.exe", has a 128-bit MD5 checksum value equal to 374394a3d46c812f5f6db425ad88f57c for the file "md5.exe". Similarly, the software enhancement program is given a MD5 checksum value with a 128-bit representation. As a result, the software enhancement is uniquely 30 marked and the MD5 checksum value 402 is used to distinguish the software enhancement file.

A cryptographic technique is applied to the cryptographic checksum by attaching a private key 408 to the software enhancement file. The private key 408 is a mathematically generated number that is unique to the sender, and is a number that only the sender knows. The software enhancement is then given a digital signature 410 which is a function of the 5 message digest number and the private key 408. The signed message is then delivered to the customer 414. The customer 414 applies an MD5 algorithm 416 to the software enhancement delivered to confirm the cryptographic checksum value 402. The recipient compares 420 the value obtained from the cryptographic checksum with the value obtained by using a (public) verification key 422. If the values are equivalent, then the 10 original PGP™ message delivered was created by the holder of the private key.

Variations, modifications, and other implementations of what is described herein will occur to those of ordinary skill in the art without departing from the spirit and the scope of the invention as claimed. Accordingly, the invention is to be defined not by the preceding illustrative description but instead by the spirit and scope of the following 15 claims.

CLAIMS:

1. A method for automatically providing software enhancements, comprising:
 - (A) providing computer security software for execution on at least one computer; and
 - 5 (B) automatically and electronically sending to the computer an enhancement to the software whenever the enhancement becomes available.
- 10 2. The method of claim 1 wherein the sending step comprises sending to the computer the enhancement which comprises an update to the software.
- 15 3. The method of claim 2 wherein the sending step comprises sending the update which comprises a database update.
4. The method of claim 2 further comprising integrating the update into the software.
- 15 5. The method of claim 4 further comprising, before the integrating step, performing at the computer a check on the update to determine its integrity and authenticity.
- 20 6. The method of claim 5 wherein the performing step comprises performing a cryptographic technique.
- 25 7. The method of claim 6 wherein the performing step comprises using a digital signature.
8. The method of claim 1 wherein the sending step comprises sending to the computer the enhancement which comprises a new version of the software.
9. The method of claim 8 further comprising installing the new version of the software on the computer.

10. The method of claim 9 further comprising, before the installing step, performing at the computer a check on the new version of the software to determine its integrity and authenticity.

5 11. The method of claim 10 wherein the performing step comprises performing a cryptographic technique..

12. The method of claim 11 wherein the performing step comprises using a digital signature.

10

13. A method for automatically providing updates, comprising:

- (A) providing computer security software for execution on at least one computer;
- (B) electronically sending to the computer an update to the software; and
- 15 (C) integrating the update into the software.

14. The method of claim 13 further comprising, before the integrating step, performing at the computer a check on the update to determine its integrity and authenticity.

20 15. The method of claim 14 wherein the performing step comprises performing a cryptographic technique.

16. The method of claim 15 wherein the performing step comprises using a digital signature.

25

17. The method of claim 13 wherein the sending step comprises automatically and electronically sending to the computer the update whenever the update becomes available.

18. The method of claim 13 wherein the sending step comprises electronically sending 30 to the computer the update upon a request.

19. The method of claim 13 wherein the sending step comprises electronically sending the update which comprises a database update.

1/7

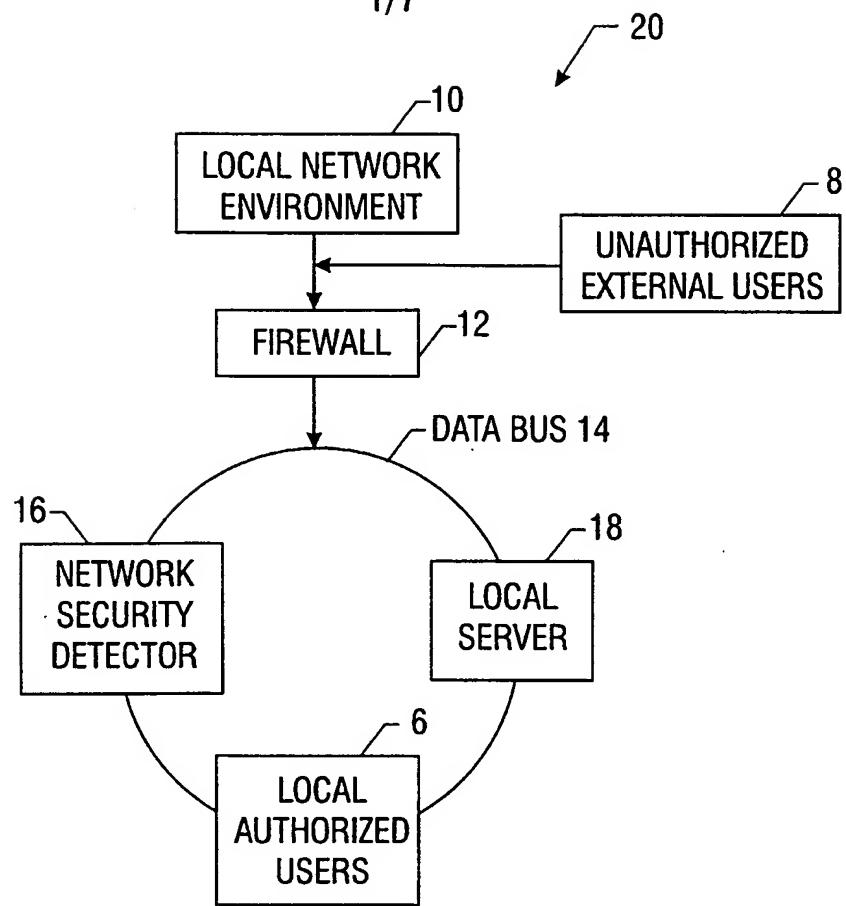


FIG. 1

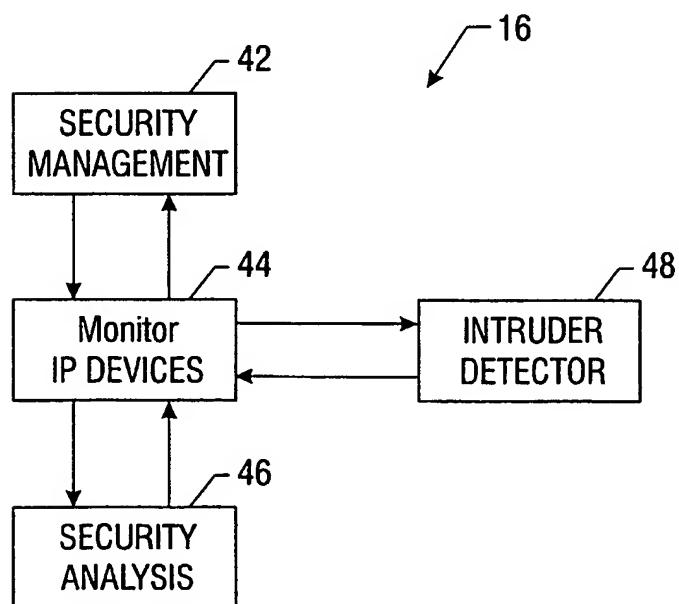


FIG. 2

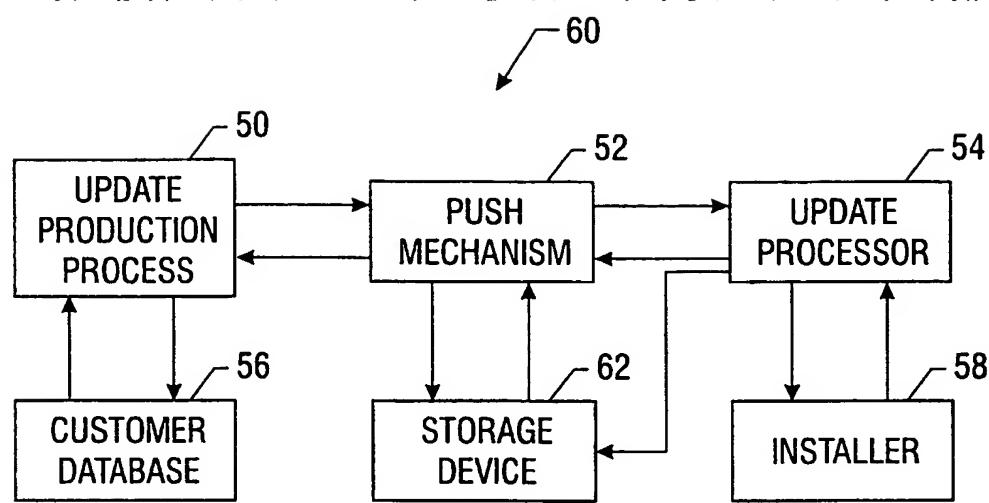


FIG. 3

3/7

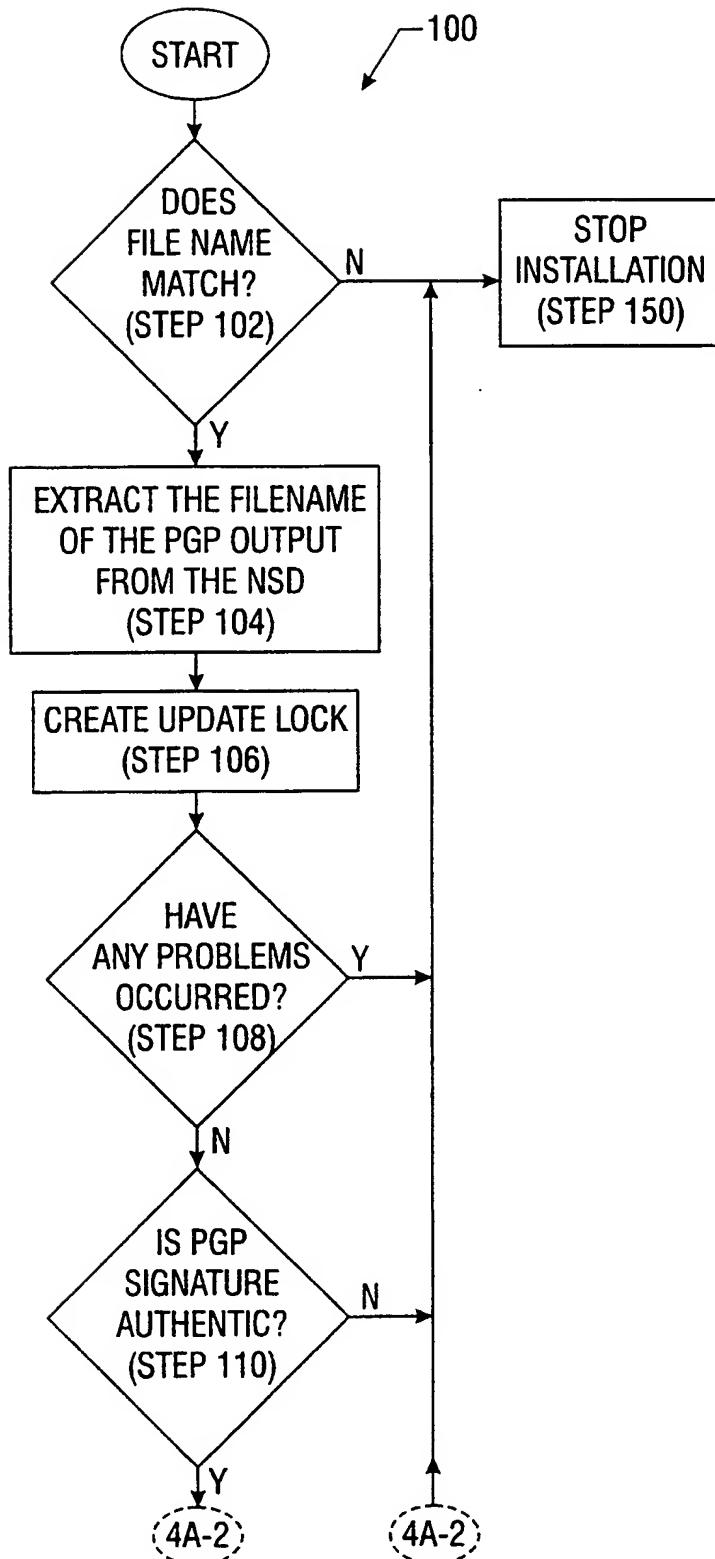


FIG. 4A-1

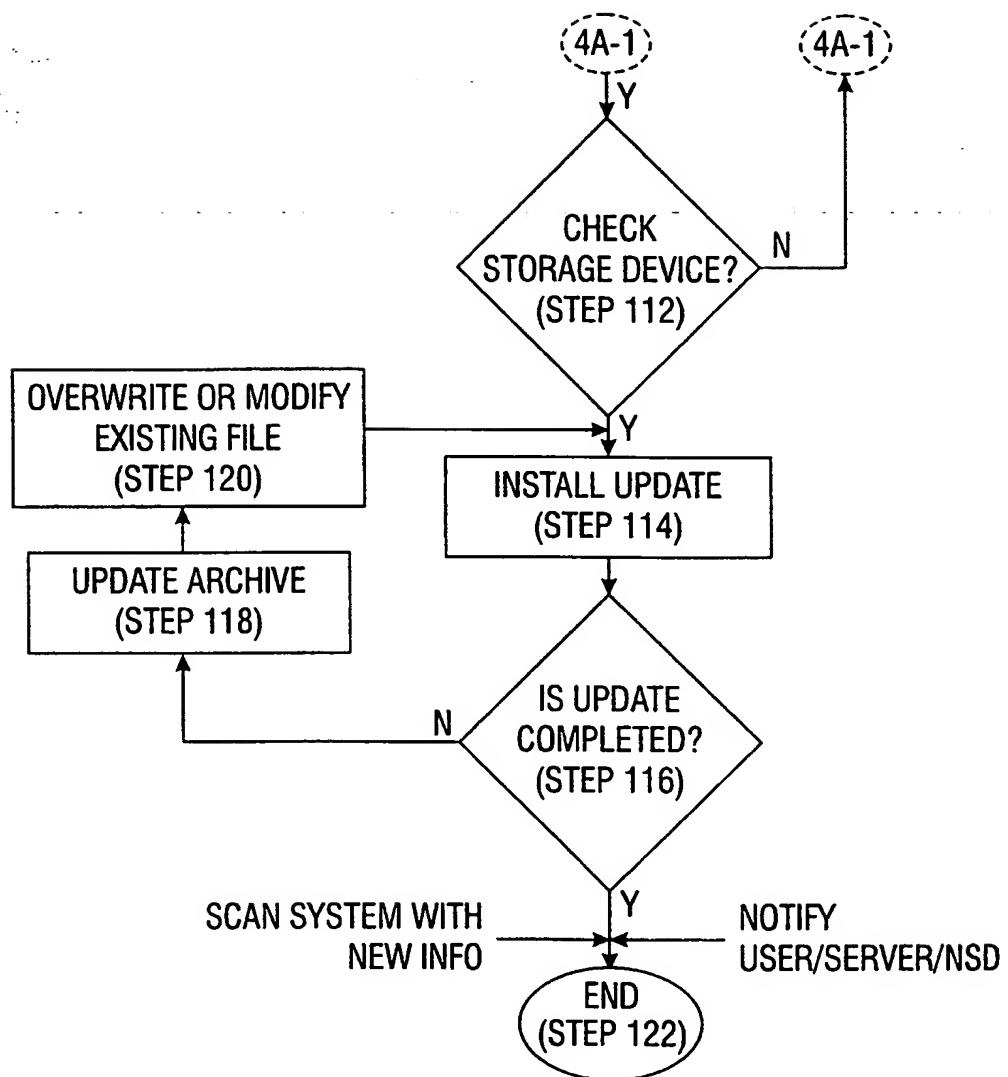


FIG. 4A-2

5/7

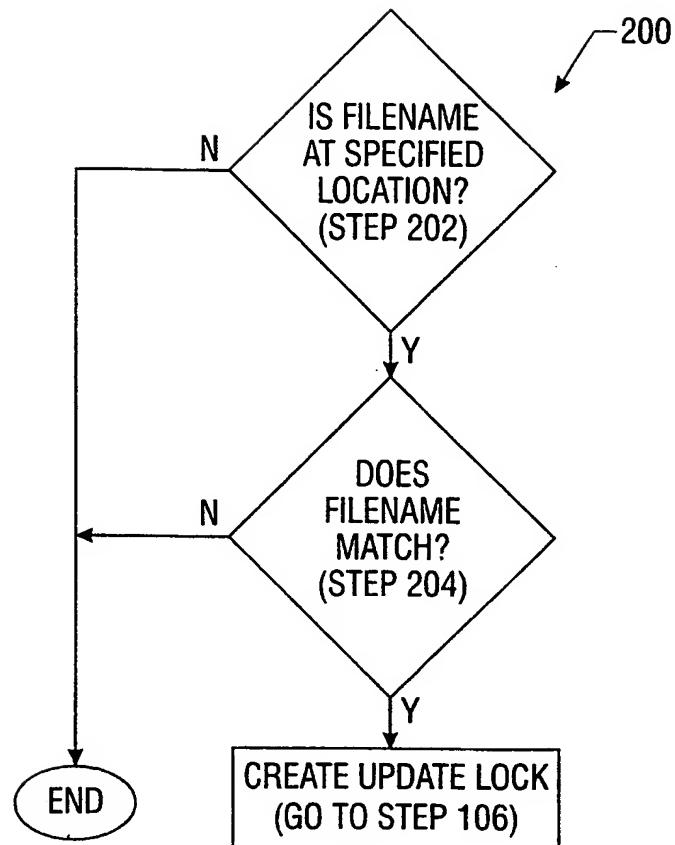


FIG. 4B

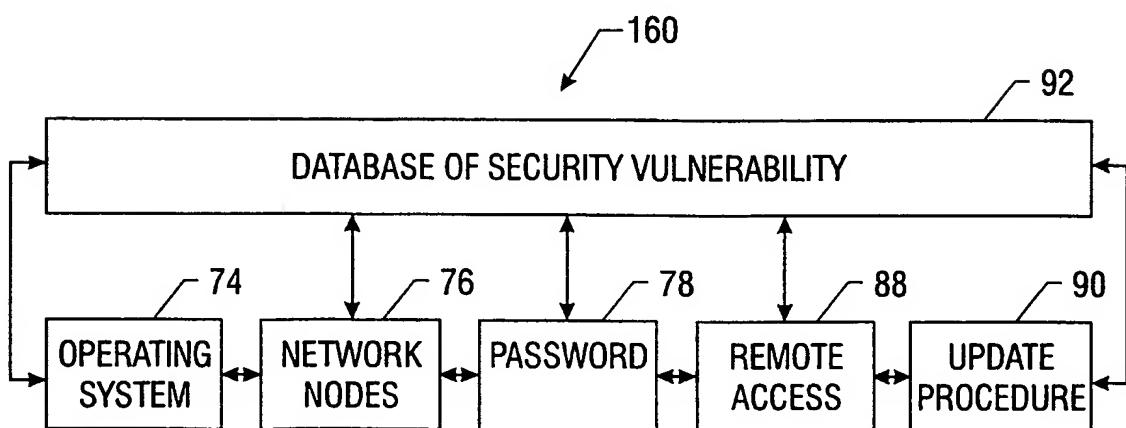


FIG. 5

SUBSTITUTE SHEET (RULE 26)

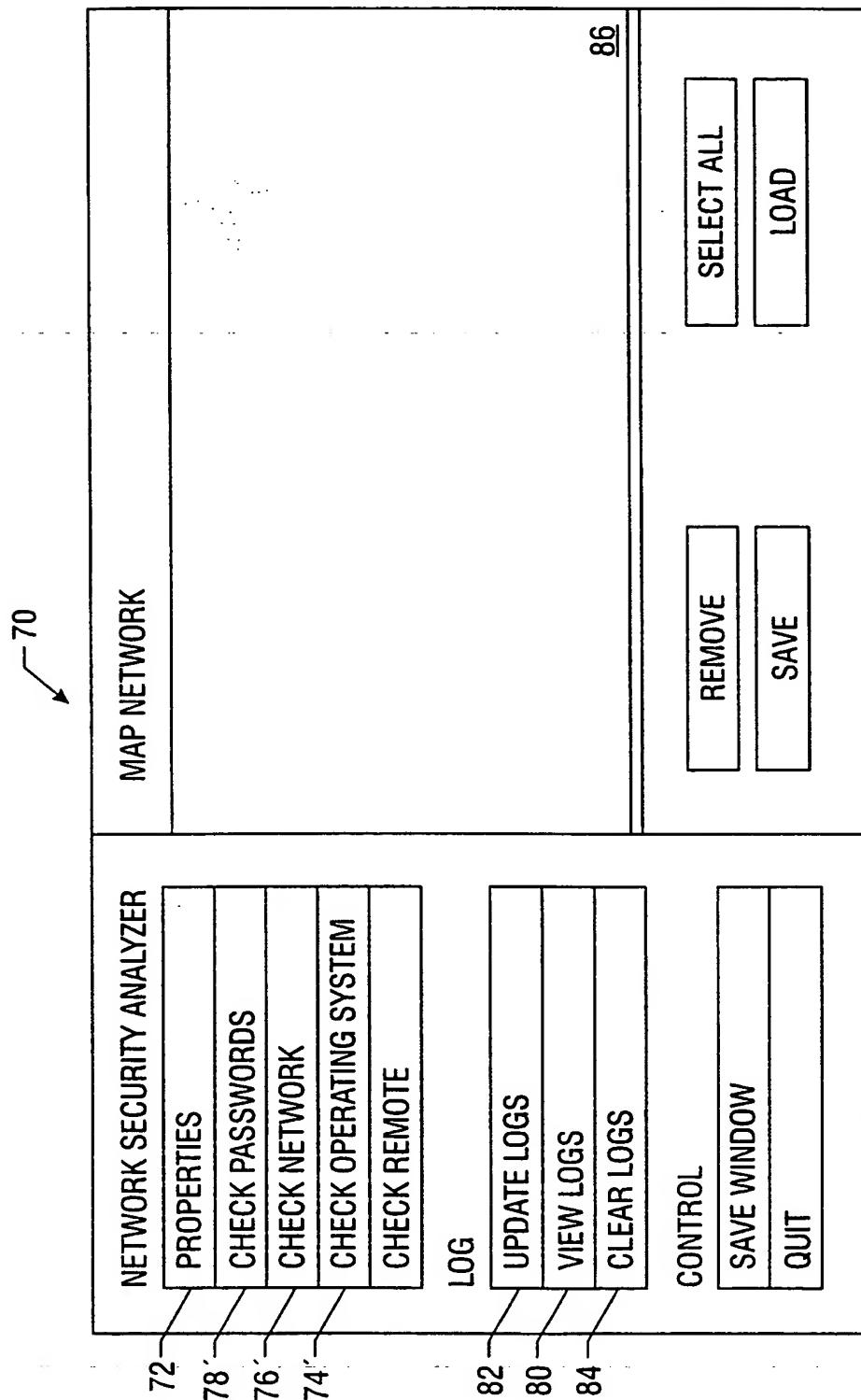


FIG. 6

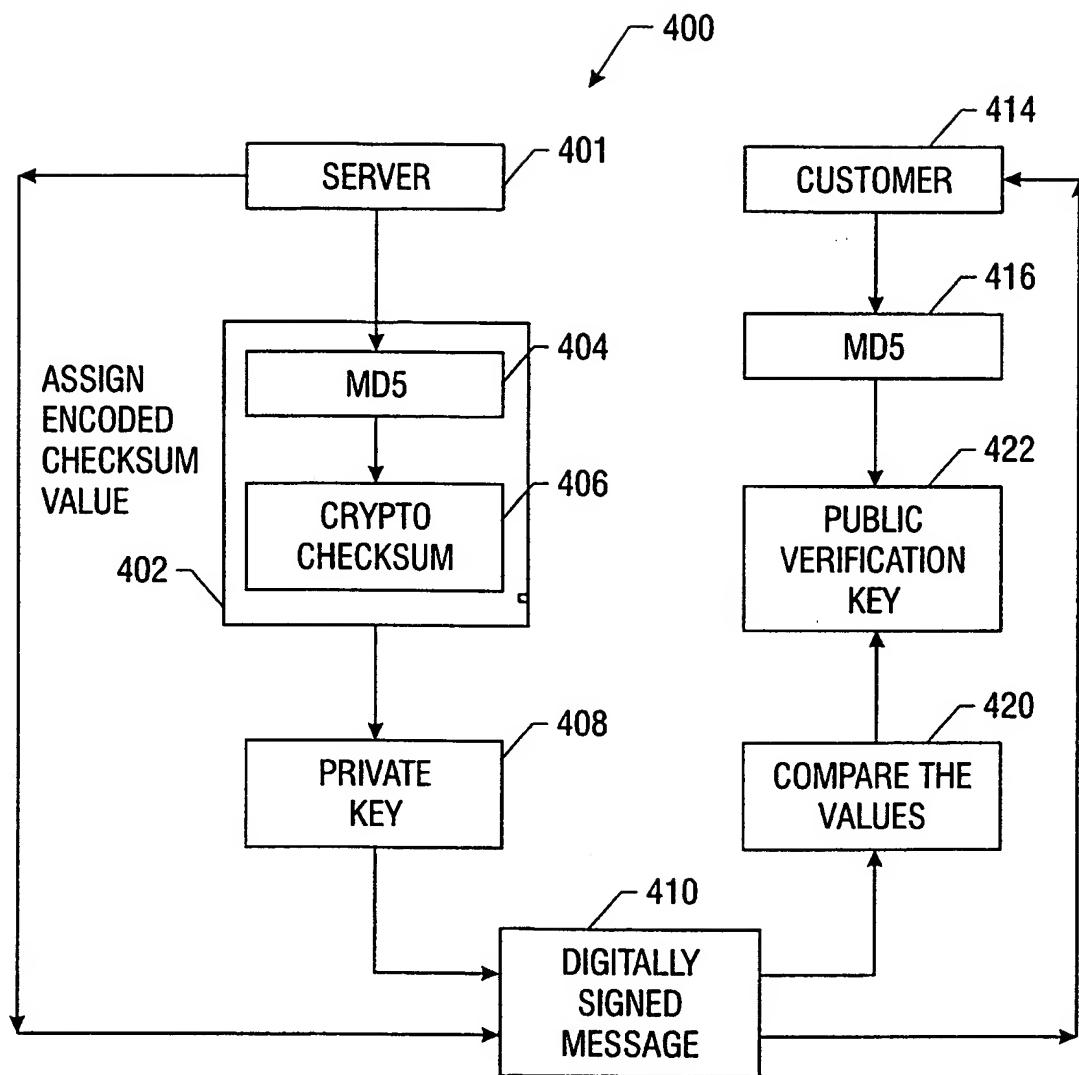


FIG. 7

INTERNATIONAL SEARCH REPORT

In. International Application No

PCT/US 99/09622

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F1/00 G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used).

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 97 04394 A (DRAKE) 6 February 1997 (1997-02-06) page 3, line 25 - line 31 page 5, line 5 - line 8 abstract; figures 3,5,6 ---	1,2,4,8, 9,13,17
A	EP 0 811 942 A (CYBER MEDIA INC) 10 December 1997 (1997-12-10) page 11, line 59 - page 12, line 14 page 17, line 9 - line 12 abstract; figures 2,9 ---	1,2, 4-16,18
A	US 5 732 275 A (KULLICK ET AL.) 24 March 1998 (1998-03-24) column 5, line 49 - column 6, line 26 column 8, line 31 - line 37 abstract; figures 4,5 ---	1-4,8,9, 13,17,19

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

Date of mailing of the international search report

13 July 1999

19/07/1999

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl
Fax: (+31-70) 340-3016

Authorized officer

Taylor, P

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 99/09622

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 782 080 A (AT&T) 2 July 1997 (1997-07-02) page 2, line 51 – page 3, line 41; figures 4,5 ---	1-5, 8-10,13, 14,17-19
A	WO 98 07085 A (SMITH ET AL.) 19 February 1998 (1998-02-19) abstract; figures 1,3 ---	1,2,4,8, 9,13,17, 18
P,A	US 5 835 911 A (NAKAGAWA ET AL.) 10 November 1998 (1998-11-10) abstract; claim 1; figures 1-6 -----	1-4,8,9, 13,18,19

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 99/09622

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
WO 9704394	A	06-02-1997	AU	5945796 A	23-01-1997
EP 811942	A	10-12-1997	AU	2477797 A	11-12-1997
			CA	2207162 A	07-12-1997
			JP	10091407 A	10-04-1998
US 5732275	A	24-03-1998		NONE	
EP 782080	A	02-07-1997	US	5706431 A	06-01-1998
			CA	2192258 A	30-06-1997
			JP	9200269 A	31-07-1997
WO 9807085	A	19-02-1998	AU	3793997 A	06-03-1998
US 5835911	A	10-11-1998	JP	7225723 A	22-08-1995
			JP	7225724 A	22-08-1995
			JP	9054734 A	25-02-1997